Weighted Last-Step Min-Max Algorithm with Improved Sub-Logarithmic Regret

Edward Moroshko[a], Koby Crammer[a]

[a] *Department of Electrical Engineering, Technion, Israel*

Abstract

In online learning the performance of an algorithm is typically compared to the performance of a fixed function from some class, with a quantity called regret. Forster [12] proposed a last-step min-max algorithm which was somewhat simpler than the algorithm of Vovk [26], yet with the same regret. In fact the algorithm he analyzed assumed that the choices of the adversary are bounded, yielding artificially only the two extreme cases. We fix this problem by weighing the examples in such a way that the min-max problem will be well defined, and provide analysis with logarithmic regret that may have better multiplicative factor than both bounds of Forster [12] and Vovk [26]. We also derive a new bound that may be sub-logarithmic, as a recent bound of Orabona et.al [21], but may have better multiplicative factor. Finally, we analyze the algorithm in a weak-type of non-stationary setting, and show a bound that is sublinear if the non-stationarity is sub-linear as well.

Keywords: Online learning, Regression, Min-max learning

1. Introduction

We consider the online learning regression problem, in which a learning algorithm tries to predict real numbers in a sequence of rounds given some side-information or inputs $\mathbf{x}_t \in \mathbb{R}^d$. Real-world example applications for these algorithms are weather or stockmarket predictions. The goal of the algorithm is to have a small discrepancy between its predictions and the associated outcomes $y_t \in \mathbb{R}$. This discrepancy is measured with a loss function, such as the square loss. It is common to evaluate algorithms by their regret, the difference between the cumulative loss of an algorithm with the cumulative loss of any function taken from some class.

Forster [12] proposed a last-step min-max algorithm for online regression that makes a prediction assuming it is the last example to be observed, and the goal of the algorithm is indeed to minimize the regret with respect to linear functions. The resulting optimization problem he obtained was convex in both choice of the algorithm and the choice of the adversary, yielding an unbounded optimization problem. Forster

Email addresses: `edward.moroshko@gmail.com` (Edward Moroshko), `koby@ee.technion.ac.il` (Koby Crammer)

circumvented this problem by assuming a bound Y over the choices of the adversary that should be known to the algorithm, yet his analysis is for the version with no bound.

We propose a modified last-step min-max algorithm with weights over examples, that are controlled in a way to obtain a problem that is concave over the choices of the adversary and convex over the choices of the algorithm. We analyze our algorithm and show a logarithmic-regret that may have a better multiplicative factor than the analysis of Forster. We derive additional analysis that is logarithmic in the loss of the reference function, rather than the number of rounds T. This behaviour was recently given by Orabona et.al [21] for a certain online-gradient decent algorithm. Yet, their bound [21] has a similar multiplicative factor to that of Forster [12], while our bound has a potentially better multiplicative factor and it has the same dependency in the cumulative loss of the reference function as Orabona et.al [21]. Additionally, our algorithm and analysis are totally free of assuming the bound Y or knowing its value.

Competing with the best *single* function might not suffice for some problems. In many real-world applications, the true target function is not fixed, but may change from time to time. We bound the performance of our algorithm also in non-stationary environment, where we measure the complexity of the non-stationary environment by the total deviation of a collection of linear functions from some fixed reference point. We show that our algorithm maintains an average loss close to that of the best sequence of functions, as long as the total of this deviation is sublinear in the number of rounds T.

A short version appeared in The 23rd International Conference on Algorithmic Learning Theory (ALT 2012). This journal version of the paper includes additionally: (1) Recursive form of the algorithm and comparison to other algorithms of the same form (Sec. 3.1). (2) Kernel version of the algorithm (Sec. 3.2). (3) MAP interpretation of the minimization problems (Remark 1 and Remark 2). (4) All proofs and extended related-work section.

2. Problem Setting

We work in the online setting for regression evaluated with the squared loss. Online algorithms work in rounds or iterations. On each iteration an online algorithm receives an instance $\mathbf{x}_t \in \mathbb{R}^d$ and predicts a real value $\hat{y}_t \in \mathbb{R}$, it then receives a label $y_t \in \mathbb{R}$, possibly chosen by an adversary, suffers loss $\ell_t(\text{alg}) = \ell\left(y_t, \hat{y}_t\right) = \left(\hat{y}_t - y_t\right)^2$, updates its prediction rule, and proceeds to the next round. The cumulative loss suffered by the algorithm over T iterations is,

$$L_T(\text{alg}) = \sum_{t=1}^{T} \ell_t(\text{alg}) \ . \tag{1}$$

The goal of the algorithm is to perform well compared to any predictor from some function class.

A common choice is to compare the performance of an algorithm with respect to *a single* function, or specifically a single linear function, $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{u}$, parameterized by a vector $\mathbf{u} \in \mathbb{R}^d$. Denote by $\ell_t(\mathbf{u}) =$

$\left(\mathbf{x}_t^\top \mathbf{u} - y_t\right)^2$ the instantaneous loss of a vector $\mathbf{u}$, and by $L_T(\mathbf{u}) = \sum_t^T \ell_t(\mathbf{u})$. The regret with respect to $\mathbf{u}$ is defined to be,

$$R_T(\mathbf{u}) = \sum_t^T (y_t - \hat{y}_t)^2 - L_T(\mathbf{u}) \ .$$

A desired goal of the algorithm is to have $R_T(\mathbf{u}) = o(T)$, that is, the average loss suffered by the algorithm will converge to the average loss of the best linear function $\mathbf{u}$.

Below in Sec. 5 we will also consider an extension of this form of regret, and evaluate the performance of an algorithm against some T-tuple of functions, $(\mathbf{u}_1, \ldots, \mathbf{u}_T) \in \mathbb{R}^d \times \cdots \times \mathbb{R}^d$,

$$R_T(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_t^T (y_t - \hat{y}_t)^2 - L_T(\mathbf{u}_1, \ldots, \mathbf{u}_T) \ ,$$

where $L_T(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_t^T \ell_t(\mathbf{u}_t)$. Clearly, with no restriction of the T-tuple, any algorithm may suffer a regret linear in T, as one can set $\mathbf{u}_t = \mathbf{x}_t(y_t / \left\|\mathbf{x}_t\right\|^2)$, and suffer zero quadratic loss in all rounds. Thus, we restrict below the possible choices of T-tuple either explicitly, or implicitly via some penalty.

3. A Last Step Min-Max Algorithm

Our algorithm is derived based on a last-step min-max prediction, proposed by Forster [12] and Takimoto and Warmuth [24]. See also the work of Azoury and Warmuth [1]. An algorithm following this approach outputs the min-max prediction assuming the current iteration is the last one. The algorithm we describe below is based on an extension of this notion. For this purpose we introduce a weighted cumulative loss using positive input-dependent weights $\{a_t\}_{t=1}^T$,

$$L_T^{\boldsymbol{a}}(\mathbf{u}) = \sum_{t=1}^T a_t \left(y_t - \mathbf{u}^\top \mathbf{x}_t\right)^2 \quad , \quad L_T^{\boldsymbol{a}}(\mathbf{u}_1, \ldots, \mathbf{u}_T) = \sum_{t=1}^T a_t \left(y_t - \mathbf{u}_t^\top \mathbf{x}_t\right)^2 \ .$$

The exact values of the weights a_t will be defined below.

Our variant of the last step min-max algorithm predicts[1]

$$\hat{y}_T = \arg\min_{\hat{y}_T} \max_{y_T} \left[\sum_{t=1}^T (y_t - \hat{y}_t)^2 - \inf_{\mathbf{u}} \left(b \left\|\mathbf{u}\right\|^2 + L_T^{\boldsymbol{a}}(\mathbf{u}) \right) \right] \ , \tag{2}$$

for some positive constant $b > 0$. We next compute the actual prediction based on the optimal last step min-max solution. We start with additional notation,

$$\mathbf{A}_t = b\mathbf{I} + \sum_{s=1}^t a_s \mathbf{x}_s \mathbf{x}_s^\top \qquad \in \mathbb{R}^{d\times d} \tag{3}$$

$$\mathbf{b}_t = \sum_{s=1}^t a_s y_s \mathbf{x}_s \qquad \in \mathbb{R}^d \ . \tag{4}$$

The solution of the internal infimum over $\mathbf{u}$ is summarized in the following lemma.

[1] y_T and $\hat{y}_T$ serves both as quantifiers (over the min and max operators, respectively), and as the optimal values over this optimization problem.

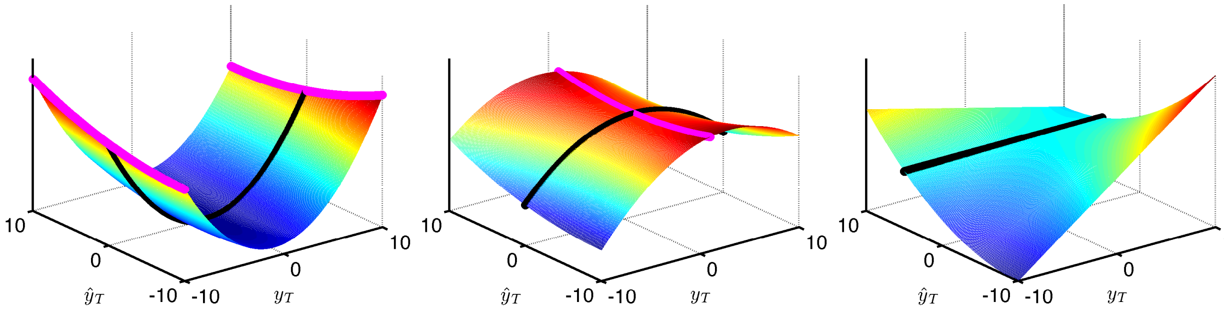

Figure 1: An illustration of the minmax objective function $G(y_T, \hat{y}_T)$ (7). The black line is the value of the objective as a function of y_T for the optimal predictor $\hat{y}_T$. Left: Forster's optimization function (convex in y_T). Center: our optimization function (strictly concave in y_T, case 1 in Theorem 2). Right: our optimization function (invariant to y_T, case 2 in Theorem 2).

Lemma 1. *For all $t \geq 1$, the function $f(\mathbf{u}) = b\left\|\mathbf{u}\right\|^2 + \sum_{s=1}^t a_s \left(y_s - \mathbf{u}^\top \mathbf{x}_s\right)^2$ is minimal at a unique point $\mathbf{u}_t$ given by,*

$$\mathbf{u}_t = \mathbf{A}_t^{-1}\mathbf{b}_t \quad \textit{and} \quad f(\mathbf{u}_t) = \sum_{s=1}^t a_s y_s^2 - \mathbf{b}_t^\top \mathbf{A}_t^{-1}\mathbf{b}_t \ . \tag{5}$$

Proof: From

$$\begin{aligned} f(\mathbf{u}) &= b\left\|\mathbf{u}\right\|^2 + \sum_{s=1}^t a_s \left(y_s - \mathbf{u}^\top \mathbf{x}_s\right)^2 \\ &= \sum_{s=1}^t a_s y_s^2 - 2\sum_{s=1}^t \mathbf{u}^\top \left(a_s y_s \mathbf{x}_s\right) + \mathbf{u}^\top \left(b\mathbf{I} + \sum_{s=1}^t a_s \mathbf{x}_s \mathbf{x}_s^\top\right)\mathbf{u} \\ &\overset{(3),(4)}{=} \sum_{s=1}^t a_s y_s^2 - 2\mathbf{u}^\top \mathbf{b}_t + \mathbf{u}^\top \mathbf{A}_t \mathbf{u} \end{aligned}$$

it follows that $\nabla f(\mathbf{u}) = 2\mathbf{A}_t\mathbf{u} - 2\mathbf{b}_t$, $\triangle f(\mathbf{u}) = 2\mathbf{A}_t$. Thus f is convex and it is minimal if $\nabla f(\mathbf{u}) = 0$, i.e. for $\mathbf{u} = \mathbf{A}_t^{-1}\mathbf{b}_t$. This show that $\mathbf{u}_t = \mathbf{A}_t^{-1}\mathbf{b}_t$ and we obtain

$$f(\mathbf{u}_t) = f\left(\mathbf{A}_t^{-1}\mathbf{b}_t\right) = \sum_{s=1}^t a_s y_s^2 - 2\mathbf{b}_t^\top \mathbf{A}_t^{-1}\mathbf{b}_t + \mathbf{b}_t^\top \mathbf{A}_t^{-1}\mathbf{A}_t\mathbf{A}_t^{-1}\mathbf{b}_t = \sum_{s=1}^t a_s y_s^2 - \mathbf{b}_t^\top \mathbf{A}_t^{-1}\mathbf{b}_t \ .$$

■

Remark 1. *The minimization problem in Lemma 1 can be interpreted as MAP estimator of $\mathbf{u}$ based on the sequence $\{(\mathbf{x}_s, y_s)\}_{s=1}^t$ in the following generative model:*

$$\begin{aligned} \mathbf{u} &\sim N\left(0, \sigma_b^2 \mathbf{I}\right) \\ y_s &\sim N\left(\mathbf{x}_s^\top \mathbf{u}, \sigma_s^2\right) \ , \end{aligned}$$

where $\sigma_b^2 = \frac{1}{2b}$ and $\sigma_s^2 = \frac{1}{2a_s}$.

Under the model we calculate,

$$\begin{aligned}
\mathbf{u}_{MAP} &= \arg\max_{\mathbf{u}} P\left(\mathbf{u} \mid \left\{\mathbf{x}_s\right\}, \left\{y_s\right\}\right) \\
&= \arg\max_{\mathbf{u}} \left[P\left(\mathbf{u}\right) \prod_{s=1}^{t} P\left(y_s \mid \mathbf{u}, \mathbf{x}_s\right) \right] \\
&= \arg\min_{\mathbf{u}} \left[-\log P\left(\mathbf{u}\right) - \sum_{s=1}^{t} \log P\left(y_s \mid \mathbf{u}, \mathbf{x}_s\right) \right] \ . \qquad (6)
\end{aligned}$$

By our gaussian generative model,

$$\begin{aligned}
-\log P\left(\mathbf{u}\right) &= \log\left(2\pi\sigma_b^2\right)^{d/2} + \frac{1}{2\sigma_b^2}\left\|\mathbf{u}\right\|^2 \\
-\log P\left(y_s \mid \mathbf{u}, \mathbf{x}_s\right) &= \log\left(2\pi\sigma_s^2\right)^{1/2} + \frac{1}{2\sigma_s^2}\left(y_s - \mathbf{x}_s^\top \mathbf{u}\right)^2 \ .
\end{aligned}$$

Substituting in (6) *we get*

$$\mathbf{u}_{MAP} = \arg\min_{\mathbf{u}} \left[\frac{1}{2\sigma_b^2}\left\|\mathbf{u}\right\|^2 + \sum_{s=1}^{t} \frac{1}{2\sigma_s^2}\left(y_s - \mathbf{x}_s^\top \mathbf{u}\right)^2 \right] \ ,$$

and by using $\frac{1}{2\sigma_b^2} = b$*,* $\frac{1}{2\sigma_s^2} = a_s$ *we get the minimization problem of Lemma 1.*

Substituting (5) back in (2) we obtain the following form of the minmax problem,

$$\min_{\hat{y}_T} \max_{y_T} G(y_T, \hat{y}_T) \quad \text{for} \quad G(y_T, \hat{y}_T) = \alpha(a_T) y_T^2 + 2\beta(a_T, \hat{y}_T) y_T + \hat{y}_T^2 \ , \qquad (7)$$

for some functions $\alpha(a_T)$ and $\beta(a_T, \hat{y}_T)$. Clearly, for this problem to be well defined the function G should be convex in $\hat{y}_T$ and concave in y_T.

A previous choice, proposed by Forster [12], is to have uniform weights and set $a_t = 1$ (for $t = 1, \ldots, T$), which for the particular function $\alpha(a_T)$ yields $\alpha(a_T) > 0$. Thus, $G(y_T, \hat{y}_T)$ is a convex function in y_T, implying that the optimal value of G is not bounded from above. Forster [12] addressed this problem by restricting y_T to belong to a predefined interval $[-Y, Y]$, known also to the learner. As a consequence, the adversary optimal prediction is in fact either $y_T = Y$ or $y_T = -Y$, which in turn yields an optimal predictor which is clipped at this bound, $\hat{y}_T = \text{clip}\left(\mathbf{b}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T, Y\right)$, where for $y > 0$ we define $\text{clip}(x, y) = x$ if $|x| \leq y$ and $\text{clip}(x, y) = y\,\text{sign}(x)$, otherwise.

This phenomena is illustrated in the left panel of Fig. 1 (best viewed in color). For the minmax optimization function defined by Forster [12], fixing some value of $\hat{y}_T$, the function is convex in y_T, and the adversary would achieve a maximal value at the boundary of the feasible values of y_T interval. That is, either $y_T = Y$ or $y_T = -Y$, as indicated by the two magenta lines at $y_T = \pm 10$. The optimal predictor $\hat{y}_T$ is achieved somewhere along the lines $y_T = Y$ or $y_T = -Y$.

We propose an alternative approach to make the minmax optimal solution bounded by appropriately setting the weight a_T such that $G(y_T, \hat{y}_T)$ is concave in y_T for a constant $\hat{y}_T$. We explicitly consider two cases. First, set a_T such that $G(y_T, \hat{y}_T)$ is *strictly concave* in y_T, and thus attains a single maximum with no need to artificially restrict the value of y_T. In this case our function is concave in y_T in the first option and has a maximum point, which is the worst adversary. The optimal predictor $\hat{y}_T$ is achieved in the unique saddle point, as illustrated in the center panel of Fig. 1. A second case is to set a_T such that $\alpha(a_T) = 0$ and the minmax function $G(y_T, \hat{y}_T)$ becomes linear in y_T. Here, the optimal prediction is achieved by choosing $\hat{y}_T$ such that $\beta(a_T, \hat{y}_T) = 0$ which turns $G(y_T, \hat{y}_T)$ to be invariant to y_T, as illustrated in the right panel of Fig. 1.

Equipped with Lemma 1 we develop the optimal solution of the min-max predictor, summarized in the following theorem.

Theorem 2. *Assume that* $1 + a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T - a_T \leq 0$. *Then the optimal prediction for the last round* T *is*

$$\hat{y}_T = \mathbf{b}_{T-1}^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T \ . \tag{8}$$

The proof of the theorem makes use of the following technical lemma.

Lemma 3. *For all* $t = 1, 2, \ldots, T$

$$a_t^2 \mathbf{x}_t^\top \mathbf{A}_t^{-1} \mathbf{x}_t + 1 - a_t = \frac{1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t - a_t}{1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t} \ . \tag{9}$$

The proof appears in Appendix A. We now prove Theorem 2.

Proof: The adversary can choose any y_T, thus the algorithm should predict $\hat{y}_T$ such that the following quantity is minimal,

$$\max_{y_T} \left(\sum_{t=1}^T (y_t - \hat{y}_t)^2 - \inf_{\mathbf{u} \in \mathbb{R}^d} \left(b \left\| \mathbf{u} \right\|^2 + \sum_{t=1}^T a_t \left(y_t - \mathbf{u}^\top \mathbf{x}_t \right)^2 \right) \right)$$
$$\overset{(5)}{=} \max_{y_T} \left(\sum_{t=1}^T (y_t - \hat{y}_t)^2 - \sum_{t=1}^T a_t y_t^2 + \mathbf{b}_T^\top \mathbf{A}_T^{-1} \mathbf{b}_T \right) \ .$$

That is, we need to solve the following minmax problem

$$\min_{\hat{y}_T} \max_{y_T} \left(\sum_{t=1}^T (y_t - \hat{y}_t)^2 - \sum_{t=1}^T a_t y_t^2 + \mathbf{b}_T^\top \mathbf{A}_T^{-1} \mathbf{b}_T \right) \ .$$

We use the following relation to re-write the optimization problem,

$$\mathbf{b}_T^\top \mathbf{A}_T^{-1} \mathbf{b}_T = \mathbf{b}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{b}_{T-1} + 2 a_T y_T \mathbf{b}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T + a_T^2 y_T^2 \mathbf{x}_T^\top \mathbf{A}_T^{-1} \mathbf{x}_T \ . \tag{10}$$

Omitting all terms that are not depending on y_T and $\hat{y}_T$,

$$\min_{\hat{y}_T} \max_{y_T} \left((y_T - \hat{y}_T)^2 - a_T y_T^2 + 2 a_T y_T \mathbf{b}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T + a_T^2 y_T^2 \mathbf{x}_T^\top \mathbf{A}_T^{-1} \mathbf{x}_T \right) \ .$$

We manipulate the last problem to be of form (7) using Lemma 3,

$$\min_{\hat{y}_T}\max_{y_T}\left(\frac{1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T-a_T}{1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T}y_T^2+2y_T\left(a_T\mathbf{b}_{T-1}^\top\mathbf{A}_T^{-1}\mathbf{x}_T-\hat{y}_T\right)+\hat{y}_T^2\right), \tag{11}$$

where

$$\alpha(a_T)=\frac{1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T-a_T}{1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T}\quad\text{and}\quad\beta(a_T,\hat{y}_T)=a_T\mathbf{b}_{T-1}^\top\mathbf{A}_T^{-1}\mathbf{x}_T-\hat{y}_T\ .$$

We consider two cases: (1) $1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T-a_T<0$ (corresponding to the middle panel of Fig. 1), and (2) $1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T-a_T=0$ (corresponding to the right panel of Fig. 1), starting with the first case,

$$1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T-a_T<0\ . \tag{12}$$

Denote the inner-maximization problem by,

$$f\left(y_T\right)=\frac{1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T-a_T}{1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T}y_T^2+2y_T\left(a_T\mathbf{b}_{T-1}^\top\mathbf{A}_T^{-1}\mathbf{x}_T-\hat{y}_T\right)+\hat{y}_T^2\ .$$

This function is strictly-concave with respect to y_T because of (12). Thus, it has a unique maximal value given by,

$$\begin{aligned}f^{max}(\hat{y}_T)\quad=\quad&-\frac{a_T}{1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T-a_T}\hat{y}_T^2+\frac{2a_T\mathbf{b}_{T-1}^\top\mathbf{A}_T^{-1}\mathbf{x}_T\left(1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T\right)}{1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T-a_T}\hat{y}_T\\&-\frac{\left(a_T\mathbf{b}_{T-1}^\top\mathbf{A}_T^{-1}\mathbf{x}_T\right)^2\left(1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T\right)}{1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T-a_T}\ .\end{aligned}$$

Next, we solve $\min_{\hat{y}_T}f^{max}(\hat{y}_T)$, which is strictly-convex with respect to $\hat{y}_T$ because of (12). Solving this problem we get the optimal last step minmax predictor,

$$\hat{y}_T=\mathbf{b}_{T-1}^\top\mathbf{A}_T^{-1}\mathbf{x}_T\left(1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T\right)\ . \tag{13}$$

We further derive the last equation. From (3) we have,

$$\mathbf{A}_T^{-1}a_T\mathbf{x}_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}=\mathbf{A}_T^{-1}\left(\mathbf{A}_T-\mathbf{A}_{T-1}\right)\mathbf{A}_{T-1}^{-1}=\mathbf{A}_{T-1}^{-1}-\mathbf{A}_T^{-1}\ . \tag{14}$$

Substituting (14) in (13) we have the following equality as desired,

$$\hat{y}_T=\mathbf{b}_{T-1}^\top\mathbf{A}_T^{-1}\mathbf{x}_T+\mathbf{b}_{T-1}^\top\mathbf{A}_T^{-1}a_T\mathbf{x}_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T=\mathbf{b}_{T-1}^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T\ . \tag{15}$$

We now move to the second case for which, $1+a_T\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T-a_T=0$, which is written equivalently as,

$$a_T=\frac{1}{1-\mathbf{x}_T^\top\mathbf{A}_{T-1}^{-1}\mathbf{x}_T}\ . \tag{16}$$

Substituting (16) in (11) we get,

$$\min_{\hat{y}_T}\max_{y_T}\left(2y_T\left(a_T\mathbf{b}_{T-1}^\top\mathbf{A}_T^{-1}\mathbf{x}_T-\hat{y}_T\right)+\hat{y}_T^2\right)\ .$$

For $\hat{y}_T \neq a_T \mathbf{b}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T$, the value of the optimization problem is not-bounded as the adversary may choose $y_T = z^2 \left(a_T \mathbf{b}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T - \hat{y}_T \right)$ for $z \to \infty$. Thus, the optimal last step minmax prediction is to set $\hat{y}_T = a_T \mathbf{b}_{T-1}^\top \mathbf{A}_T^{-1} \mathbf{x}_T$. Substituting $a_T = 1 + a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T$ and following the derivation from (13) to (15) above, yields the desired identity. ■

We conclude by noting that although we did not restrict the form of the predictor $\hat{y}_T$, it turns out that it is a linear predictor defined by $\hat{y}_T = \mathbf{x}_T^\top \boldsymbol{w}_{T-1}$ for $\boldsymbol{w}_{T-1} = \mathbf{A}_{T-1}^{-1} \mathbf{b}_{T-1}$. In other words, the functional form of the optimal predictor is the same as the form of the comparison function class - linear functions in our case. We call the algorithm (defined using (3), (4) and (8)) WEMM for weighted min-max prediction. We note that WEMM can also be seen as an incremental off-line algorithm [1] or follow-the-leader, on a weighted sequence. The prediction $\hat{y}_T = \mathbf{x}_T^\top \boldsymbol{w}_{T-1}$ is with a model that is optimal over a prefix of length $T-1$. The prediction of the optimal predictor defined in (5) is $\mathbf{x}_T^\top \mathbf{u}_{T-1} = \mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{b}_{T-1} = \hat{y}_T$, where $\hat{y}_T$ was defined in (8).

3.1. Recursive form

Although Theorem 2 is correct for $1 + a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1} \mathbf{x}_T - a_T \leq 0$, in the rest of the paper we will (almost always) assume an equality, that is

$$a_t = \frac{1}{1 - \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t} \quad , \quad t = 1 \ldots T \ . \tag{17}$$

For this case, WEMM algorithm can be expressed in a recursive form in terms of weight vector $\mathbf{w}_t$ and a covariance-like matrix $\mathbf{\Sigma}_t$. We denote $\mathbf{w}_t = \mathbf{A}_t^{-1} \mathbf{b}_t$ and $\mathbf{\Sigma}_t = \mathbf{A}_t^{-1}$, and develop recursive update rules for $\mathbf{w}_t$ and $\mathbf{\Sigma}_t$:

$$\begin{aligned}
\mathbf{w}_t &= \mathbf{A}_t^{-1} \mathbf{b}_t \\
&= \left(\mathbf{A}_{t-1} + a_t \mathbf{x}_t \mathbf{x}_t^\top \right)^{-1} \left(\mathbf{b}_{t-1} + a_t y_t \mathbf{x}_t \right) \\
&= \left(\mathbf{A}_{t-1}^{-1} - \frac{\mathbf{A}_{t-1}^{-1} \mathbf{x}_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}}{a_t^{-1} + \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t} \right) \left(\mathbf{b}_{t-1} + a_t y_t \mathbf{x}_t \right) \\
&= \mathbf{w}_{t-1} - \frac{\mathbf{A}_{t-1}^{-1} \mathbf{x}_t \mathbf{x}_t^\top \mathbf{w}_{t-1}}{a_t^{-1} + \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t} + a_t y_t \mathbf{A}_{t-1}^{-1} \mathbf{x}_t - \frac{a_t y_t \mathbf{A}_{t-1}^{-1} \mathbf{x}_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t}{a_t^{-1} + \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t} \\
&= \mathbf{w}_{t-1} + \frac{y_t \mathbf{A}_{t-1}^{-1} \mathbf{x}_t - \mathbf{A}_{t-1}^{-1} \mathbf{x}_t \mathbf{x}_t^\top \mathbf{w}_{t-1}}{a_t^{-1} + \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t} \\
&\overset{(17)}{=} \mathbf{w}_{t-1} + \left(y_t - \mathbf{x}_t^\top \mathbf{w}_{t-1} \right) \mathbf{A}_{t-1}^{-1} \mathbf{x}_t \\
&= \mathbf{w}_{t-1} + \left(y_t - \mathbf{x}_t^\top \mathbf{w}_{t-1} \right) \mathbf{\Sigma}_{t-1} \mathbf{x}_t \ ,
\end{aligned} \tag{18}$$

and

$$\begin{aligned}\mathbf{\Sigma}_t^{-1} &= \mathbf{A}_t = \mathbf{A}_{t-1} + a_t \mathbf{x}_t \mathbf{x}_t^\top \\ &\overset{(17)}{=} \mathbf{A}_{t-1} + \frac{\mathbf{x}_t \mathbf{x}_t^\top}{1 - \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t} \\ &= \mathbf{\Sigma}_{t-1}^{-1} + \frac{\mathbf{x}_t \mathbf{x}_t^\top}{1 - \mathbf{x}_t^\top \mathbf{\Sigma}_{t-1} \mathbf{x}_t}\end{aligned}$$

or

$$\mathbf{\Sigma}_t = \mathbf{\Sigma}_{t-1} - \mathbf{\Sigma}_{t-1} \mathbf{x}_t \mathbf{x}_t^\top \mathbf{\Sigma}_{t-1} \ . \qquad (19)$$

A summary of the algorithm in a recursive form appears in the right column of Table 1.

It is instructive to compare similar second order online algorithms for regression. The ridge-regression [13], summarized in the third column of Table 1, uses the previous examples to generate a weight-vector, which is used to predict current example. On round t it sets a weight-vector to be the solution of the following optimization problem,

$$\mathbf{w}_{t-1} = \arg\min_{\mathbf{w}} \left[\sum_{i=1}^{t-1} \left(y_i - \mathbf{x}_i^\top \mathbf{w} \right)^2 + b \left\| \mathbf{w} \right\|^2 \right] \ ,$$

and outputs a prediction $\hat{y}_t = \mathbf{x}_t^\top \mathbf{w}_{t-1}$. The recursive least squares (RLS) [15] is a similar algorithm, yet it uses a forgetting factor $0 < r \leq 1$, and sets the weight-vector according to

$$\mathbf{w}_{t-1} = \arg\min_{\mathbf{w}} \left[\sum_{i=1}^{t-1} r^{t-i-1} \left(y_i - \mathbf{x}_i^\top \mathbf{w} \right)^2 \right] \ .$$

The Aggregating Algorithm for regression (AAR) [27], summarized in the second column of Table 1, was introduced by Vovk and it is similar to ridge-regression, except it contains additional regularization, which eventually makes it shrink the predictions. It is an application of the Aggregating Algorithm [26] (a general algorithm for merging prediction strategies) to the problem of linear regression with square loss. On round t, the weight-vector is obtained according to

$$\mathbf{w}_t = \arg\min_{\mathbf{w}} \left[\sum_{i=1}^{t-1} \left(y_i - \mathbf{x}_i^\top \mathbf{w} \right)^2 + \left(\mathbf{x}_t^\top \mathbf{w} \right)^2 + b \left\| \mathbf{w} \right\|^2 \right] \ ,$$

and the algorithm predicts $\hat{y}_t = \mathbf{x}_t^\top \mathbf{w}_t$. Compared to ridge-regression, the AAR algorithm uses an additional input pair $(\mathbf{x}_t, 0)$. The AAR algorithm was shown to be last-step min-max optimal by Forster [12], that is the predictions can be obtained by solving (2) for $a_t = 1, \ t = 1, \ldots, T$.

The AROWR algorithm [9, 25], summarized in the left column of Table 1, is a modification of the AROW algorithm [8] for regression. It maintains a Gaussian distribution parameterized by a mean $\mathbf{w}_t \in \mathbb{R}^d$ and a full covariance matrix $\mathbf{\Sigma}_t \in \mathbb{R}^{d \times d}$. Intuitively, the mean $\mathbf{w}_t$ represents a current linear function, while the covariance matrix $\mathbf{\Sigma}_t$ captures the uncertainty in the linear function $\mathbf{w}_t$. Given a new example $(\mathbf{x}_t, y_t)$ the algorithm uses its current mean to make a prediction $\hat{y}_t = \mathbf{x}_t^\top \mathbf{w}_{t-1}$. AROWR then sets the new distribution to be the solution of the following optimization problem,

$$\arg\min_{\mathbf{w},\boldsymbol{\Sigma}} \left[\mathrm{D}_{\mathrm{KL}} \left(\mathcal{N}\left(\mathbf{w}, \boldsymbol{\Sigma}\right) \| \mathcal{N}\left(\mathbf{w}_{t-1}, \boldsymbol{\Sigma}_{t-1}\right)\right) + \frac{1}{2r}\left(y_t - \mathbf{w}^\top \mathbf{x}_t\right)^2 + \frac{1}{2r}\left(\mathbf{x}_t^\top \boldsymbol{\Sigma} \mathbf{x}_t\right)\right] .$$

Crammer et.al. [9] derived regret bounds for this algorithm.

Comparing WEMM to other algorithms we note two differences. First, for the weight-vector update rule, we do not have the normalization term $1 + \mathbf{x}_t^\top \boldsymbol{\Sigma}_{t-1}\mathbf{x}_t$. Second, for the covariance matrix update rule, our algorithm gives non-constant scale to the increment by $\mathbf{x}_t \mathbf{x}_t^\top$. This scale $1/(1 - \mathbf{x}_t^\top \boldsymbol{\Sigma}_{t-1}\mathbf{x}_t)$ is small when the current instance $\mathbf{x}_t$ lies along the directions spanned by previously observed inputs $\{\mathbf{x}_i\}_{i=1}^{t-1}$, and large when the current instance $\mathbf{x}_t$ lies along previously unobserved directions.

		AROWR [9, 25]	**AAR** [27] / **Min-Max** [12]	**Ridge-Regression** [13]	**WEMM** this work
Parameters		$0 < r, b$	$0 < b$	$0 < b$	$1 < b$
Initialize		$\mathbf{w}_0 = \mathbf{0}$, $\boldsymbol{\Sigma}_0 = b^{-1}\mathbf{I}$			
For $t = 1...T$		Receive an instance $\mathbf{x}_t$			
	Output prediction	$\hat{y}_t = \mathbf{x}_t^\top \mathbf{w}_{t-1}$	$\hat{y}_t = \frac{\mathbf{x}_t^\top \mathbf{w}_{t-1}}{1 + \mathbf{x}_t^\top \boldsymbol{\Sigma}_{t-1}\mathbf{x}_t}$	$\hat{y}_t = \mathbf{x}_t^\top \mathbf{w}_{t-1}$	$\hat{y}_t = \mathbf{x}_t^\top \mathbf{w}_{t-1}$
		Receive a correct label y_t			
	Update $\boldsymbol{\Sigma}_t$:	$\boldsymbol{\Sigma}_t^{-1} =$ $\boldsymbol{\Sigma}_{t-1}^{-1} + \frac{1}{r}\mathbf{x}_t\mathbf{x}_t^\top$	$\boldsymbol{\Sigma}_t^{-1} =$ $\boldsymbol{\Sigma}_{t-1}^{-1} + \mathbf{x}_t\mathbf{x}_t^\top$	$\boldsymbol{\Sigma}_t^{-1} =$ $\boldsymbol{\Sigma}_{t-1}^{-1} + \mathbf{x}_t\mathbf{x}_t^\top$	$\boldsymbol{\Sigma}_t^{-1} =$ $\boldsymbol{\Sigma}_{t-1}^{-1} + \frac{\mathbf{x}_t\mathbf{x}_t^\top}{1 - \mathbf{x}_t^\top \boldsymbol{\Sigma}_{t-1}\mathbf{x}_t}$
	Update $\mathbf{w}_t$:	$\mathbf{w}_t = \mathbf{w}_{t-1}$ $+ \frac{\left(y_t - \mathbf{x}_t^\top \mathbf{w}_{t-1}\right)\boldsymbol{\Sigma}_{t-1}\mathbf{x}_t}{r + \mathbf{x}_t^\top \boldsymbol{\Sigma}_{t-1}\mathbf{x}_t}$	$\mathbf{w}_t = \mathbf{w}_{t-1}$ $+ \frac{\left(y_t - \mathbf{x}_t^\top \mathbf{w}_{t-1}\right)\boldsymbol{\Sigma}_{t-1}\mathbf{x}_t}{1 + \mathbf{x}_t^\top \boldsymbol{\Sigma}_{t-1}\mathbf{x}_t}$	$\mathbf{w}_t = \mathbf{w}_{t-1}$ $+ \frac{\left(y_t - \mathbf{x}_t^\top \mathbf{w}_{t-1}\right)\boldsymbol{\Sigma}_{t-1}\mathbf{x}_t}{1 + \mathbf{x}_t^\top \boldsymbol{\Sigma}_{t-1}\mathbf{x}_t}$	$\mathbf{w}_t = \mathbf{w}_{t-1}$ $+ \left(y_t - \mathbf{x}_t^\top \mathbf{w}_{t-1}\right) \boldsymbol{\Sigma}_{t-1}\mathbf{x}_t$
Output		$\mathbf{w}_T$, $\boldsymbol{\Sigma}_T$			

Table 1: Second order online algorithms for regression

3.2. Kernel version of the algorithm

In this section we show that the WEMM algorithm can be expressed in dual variables, which allows an efficient run of the algorithm in any reproducing kernel Hilbert space. We show by induction that the weight-vector $\mathbf{w}_t$ and the covariance matrix $\boldsymbol{\Sigma}_t$ computed by the WEMM algorithm in the right column of Table 1 can be written in the form

$$\begin{aligned}\mathbf{w}_t &= \sum_{i=1}^{t} \alpha_i^{(t)} \mathbf{x}_i \\ \boldsymbol{\Sigma}_t &= \sum_{j=1}^{t}\sum_{k=1}^{t} \beta_{j,k}^{(t)} \mathbf{x}_j \mathbf{x}_k^\top + b^{-1}\mathbf{I} \ ,\end{aligned}$$

where the coefficients α_i and $\beta_{j,k}$ depend only on inner products of the input vectors.

For the initial step we have $\mathbf{w}_0 = \mathbf{0}$ and $\boldsymbol{\Sigma}_0 = b^{-1}\mathbf{I}$ which are trivially written in the desired form by setting $\alpha^{(0)} = 0$ and $\beta^{(0)} = 0$. We proceed to the induction step. From the weight-vector update rule (18) we get

$$\begin{aligned}\mathbf{w}_t &= \mathbf{w}_{t-1} + \left(y_t - \mathbf{x}_t^\top \mathbf{w}_{t-1}\right) \boldsymbol{\Sigma}_{t-1}\mathbf{x}_t = \\ &= \sum_{i=1}^{t-1} \alpha_i^{(t-1)} \mathbf{x}_i + \left(y_t - \mathbf{x}_t^\top \sum_{i=1}^{t-1} \alpha_i^{(t-1)} \mathbf{x}_i\right) \left(\sum_{j=1}^{t-1}\sum_{k=1}^{t-1} \beta_{j,k}^{(t-1)} \mathbf{x}_j \mathbf{x}_k^\top + b^{-1}\mathbf{I}\right) \mathbf{x}_t \\ &= \sum_{i=1}^{t-1} \alpha_i^{(t-1)} \mathbf{x}_i + \left(y_t - \sum_{i=1}^{t-1} \alpha_i^{(t-1)} \left(\mathbf{x}_t^\top \mathbf{x}_i\right)\right) \left(\sum_{j=1}^{t-1}\sum_{k=1}^{t-1} \beta_{j,k}^{(t-1)} \left(\mathbf{x}_k^\top \mathbf{x}_t\right) \mathbf{x}_j + b^{-1}\mathbf{x}_t\right) \\ &= \sum_{i=1}^{t-1} \left[\alpha_i^{(t-1)} + \left(y_t - \sum_{l=1}^{t-1} \alpha_l^{(t-1)} \left(\mathbf{x}_t^\top \mathbf{x}_l\right)\right) \sum_{k=1}^{t-1} \beta_{i,k}^{(t-1)} \left(\mathbf{x}_k^\top \mathbf{x}_t\right)\right] \mathbf{x}_i + b^{-1} \left(y_t - \sum_{i=1}^{t-1} \alpha_i^{(t-1)} \left(\mathbf{x}_t^\top \mathbf{x}_i\right)\right) \mathbf{x}_t \ ,\end{aligned}$$

thus

$$\alpha_i^{(t)} = \begin{cases} \alpha_i^{(t-1)} + \left(y_t - \sum_{l=1}^{t-1} \alpha_l^{(t-1)} \left(\mathbf{x}_t^\top \mathbf{x}_l\right)\right) \sum_{k=1}^{t-1} \beta_{i,k}^{(t-1)} \left(\mathbf{x}_k^\top \mathbf{x}_t\right) & i = 1, \ldots, t-1 \\ b^{-1} \left(y_t - \sum_{l=1}^{t-1} \alpha_l^{(t-1)} \left(\mathbf{x}_t^\top \mathbf{x}_l\right)\right) & i = t \end{cases}$$

From the covariance matrix update rule (19) we get

$$
\begin{aligned}
\boldsymbol{\Sigma}_t &= \boldsymbol{\Sigma}_{t-1} - \boldsymbol{\Sigma}_{t-1}\mathbf{x}_t\mathbf{x}_t^\top\boldsymbol{\Sigma}_{t-1} \\
&= \sum_{j=1}^{t-1}\sum_{k=1}^{t-1}\beta_{j,k}^{(t-1)}\mathbf{x}_j\mathbf{x}_k^\top + b^{-1}\mathbf{I} - \left(\sum_{j=1}^{t-1}\sum_{k=1}^{t-1}\beta_{j,k}^{(t-1)}\mathbf{x}_j\mathbf{x}_k^\top + b^{-1}\mathbf{I}\right)\mathbf{x}_t\mathbf{x}_t^\top\left(\sum_{j=1}^{t-1}\sum_{k=1}^{t-1}\beta_{j,k}^{(t-1)}\mathbf{x}_j\mathbf{x}_k^\top + b^{-1}\mathbf{I}\right) \\
&= \sum_{j=1}^{t-1}\sum_{k=1}^{t-1}\beta_{j,k}^{(t-1)}\mathbf{x}_j\mathbf{x}_k^\top + b^{-1}\mathbf{I} - \left(\sum_{j=1}^{t-1}\sum_{k=1}^{t-1}\beta_{j,k}^{(t-1)}\left(\mathbf{x}_k^\top\mathbf{x}_t\right)\mathbf{x}_j + b^{-1}\mathbf{x}_t\right)\left(\sum_{j=1}^{t-1}\sum_{k=1}^{t-1}\beta_{j,k}^{(t-1)}\left(\mathbf{x}_t^\top\mathbf{x}_j\right)\mathbf{x}_k^\top + b^{-1}\mathbf{x}_t^\top\right) \\
&= \sum_{j=1}^{t-1}\sum_{k=1}^{t-1}\beta_{j,k}^{(t-1)}\mathbf{x}_j\mathbf{x}_k^\top + b^{-1}\mathbf{I} - \sum_{j=1}^{t-1}\sum_{k=1}^{t-1}\sum_{l=1}^{t-1}\sum_{m=1}^{t-1}\beta_{l,m}^{(t-1)}\beta_{j,k}^{(t-1)}\left(\mathbf{x}_k^\top\mathbf{x}_t\right)\left(\mathbf{x}_t^\top\mathbf{x}_l\right)\mathbf{x}_j\mathbf{x}_m^\top \\
&\quad -b^{-1}\sum_{j=1}^{t-1}\sum_{k=1}^{t-1}\beta_{j,k}^{(t-1)}\left(\mathbf{x}_t^\top\mathbf{x}_j\right)\mathbf{x}_t\mathbf{x}_k^\top - b^{-1}\sum_{j=1}^{t-1}\sum_{k=1}^{t-1}\beta_{j,k}^{(t-1)}\left(\mathbf{x}_k^\top\mathbf{x}_t\right)\mathbf{x}_j\mathbf{x}_t^\top - b^{-2}\mathbf{x}_t\mathbf{x}_t^\top \\
&= \sum_{j=1}^{t-1}\sum_{k=1}^{t-1}\left[\beta_{j,k}^{(t-1)} - \sum_{l=1}^{t-1}\sum_{m=1}^{t-1}\beta_{l,k}^{(t-1)}\beta_{j,m}^{(t-1)}\left(\mathbf{x}_m^\top\mathbf{x}_t\right)\left(\mathbf{x}_t^\top\mathbf{x}_l\right)\right]\mathbf{x}_j\mathbf{x}_k^\top + b^{-1}\mathbf{I} \\
&\quad -b^{-1}\sum_{k=1}^{t-1}\sum_{j=1}^{t-1}\beta_{j,k}^{(t-1)}\left(\mathbf{x}_t^\top\mathbf{x}_j\right)\mathbf{x}_t\mathbf{x}_k^\top - b^{-1}\sum_{j=1}^{t-1}\sum_{k=1}^{t-1}\beta_{j,k}^{(t-1)}\left(\mathbf{x}_k^\top\mathbf{x}_t\right)\mathbf{x}_j\mathbf{x}_t^\top - b^{-2}\mathbf{x}_t\mathbf{x}_t^\top \ ,
\end{aligned}
$$

thus

$$
\beta_{j,k}^{(t)} = \begin{cases}
\beta_{j,k}^{(t-1)} - \sum_{l=1}^{t-1}\sum_{m=1}^{t-1}\beta_{l,k}^{(t-1)}\beta_{j,m}^{(t-1)}\left(\mathbf{x}_m^\top\mathbf{x}_t\right)\left(\mathbf{x}_t^\top\mathbf{x}_l\right) & j,k = 1,\ldots,t-1 \\
-b^{-1}\sum_{l=1}^{t-1}\beta_{l,k}^{(t-1)}\left(\mathbf{x}_t^\top\mathbf{x}_l\right) & j = t\ ,\ k = 1,\ldots,t-1 \\
-b^{-1}\sum_{l=1}^{t-1}\beta_{j,l}^{(t-1)}\left(\mathbf{x}_l^\top\mathbf{x}_t\right) & k = t\ ,\ j = 1,\ldots,t-1 \\
-b^{-2} & j = k = t
\end{cases}
$$

A summary of the kernel version of the `WEMM` algorithm appears in Fig. 2.

4. Analysis

We analyze the algorithm in two steps. First, in Theorem 4 we show that the algorithm suffers a *constant* regret compared with the optimal weight vector $\mathbf{u}$ evaluated using *the weighted* loss, $L^{\boldsymbol{a}}(\mathbf{u})$. Second, in Theorem 5 and Theorem 6 we show that the difference of the weighted-loss $L^{\boldsymbol{a}}(\mathbf{u})$ to the true loss $L(\mathbf{u})$ is only logarithmic in T or in $L_T(\mathbf{u})$.

Theorem 4. *Assume $1 + a_t\mathbf{x}_t^\top\mathbf{A}_{t-1}^{-1}\mathbf{x}_t - a_t \leq 0$ for $t = 1\ldots T$ (which is satisfied by our choice later). Then, the loss of `WEMM`, $\hat{y}_t = \mathbf{b}_{t-1}^\top\mathbf{A}_{t-1}^{-1}\mathbf{x}_t$ for $t = 1\ldots T$, is upper bounded by,*

$$
L_T(\texttt{WEMM}) \leq \inf_{\mathbf{u}\in\mathbb{R}^d}\left(b\left\|\mathbf{u}\right\|^2 + L_T^{\boldsymbol{a}}(\mathbf{u})\right) \ .
$$

Furthermore, if $1 + a_t\mathbf{x}_t^\top\mathbf{A}_{t-1}^{-1}\mathbf{x}_t - a_t = 0$, then the last inequality is in fact an equality.

```
Parameter:. $1 < b$, kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$
Initialize:. Set $\alpha^{(0)} = 0$ and $\beta^{(0)} = 0$
For $t = 1, \ldots, T$ do
    • Receive an instance $\mathbf{x}_t$
    • Output prediction $\hat{y}_t = \sum_{i=1}^{t-1} \alpha_i^{(t-1)} K(\mathbf{x}_t, \mathbf{x}_i)$
    • Receive the correct label $y_t$
    • Update:
        $\alpha_i^{(t)} = \begin{cases} \alpha_i^{(t-1)} + \left(y_t - \sum_{l=1}^{t-1} \alpha_l^{(t-1)} K(\mathbf{x}_t, \mathbf{x}_l)\right) \sum_{k=1}^{t-1} \beta_{i,k}^{(t-1)} K(\mathbf{x}_k, \mathbf{x}_t) & i = 1, \ldots, t-1 \\ b^{-1}\left(y_t - \sum_{l=1}^{t-1} \alpha_l^{(t-1)} K(\mathbf{x}_t, \mathbf{x}_l)\right) & i = t \end{cases} \quad (20)$
        $\beta_{j,k}^{(t)} = \begin{cases} \beta_{j,k}^{(t-1)} - \sum_{l=1}^{t-1} \sum_{m=1}^{t-1} \beta_{l,k}^{(t-1)} \beta_{j,m}^{(t-1)} K(\mathbf{x}_m, \mathbf{x}_t) K(\mathbf{x}_t, \mathbf{x}_l) & j, k = 1, \ldots, t-1 \\ -b^{-1} \sum_{l=1}^{t-1} \beta_{l,k}^{(t-1)} K(\mathbf{x}_t, \mathbf{x}_l) & j = t\ ,\ k = 1, \ldots, t-1 \\ -b^{-1} \sum_{l=1}^{t-1} \beta_{j,l}^{(t-1)} K(\mathbf{x}_l, \mathbf{x}_t) & k = t\ ,\ j = 1, \ldots, t-1 \\ -b^{-2} & j = k = t \end{cases} \quad (21)$
Output:. $\left\{\alpha_i^{(T)}\right\}_{i=1}^{T}, \left\{\beta_{j,k}^{(T)}\right\}_{j,k=1}^{T}$
```

Figure 2: Kernel WEMM

Proof sketch: Long algebraic manipulation given in Appendix B yields,

$$\ell_t(\texttt{WEMM}) + \inf_{\mathbf{u}\in\mathbb{R}^d} \left(b\|\mathbf{u}\|^2 + L_{t-1}^{\boldsymbol{a}}(\mathbf{u})\right) - \inf_{\mathbf{u}\in\mathbb{R}^d} \left(b\|\mathbf{u}\|^2 + L_t^{\boldsymbol{a}}(\mathbf{u})\right) = \frac{1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t - a_t}{1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t} (y_t - \hat{y}_t)^2 \le 0 \ .$$

Summing over t gives the desired bound. ■

Next we decompose the weighted loss $L_T^{\boldsymbol{a}}(\mathbf{u})$ into a sum of the actual loss $L_T(\mathbf{u})$ and a logarithmic term. We give two bounds - one is logarithmic in T (Theorem 5), and the second is logarithmic in $L_T(\mathbf{u})$ (Theorem 6). We use the following notation of the loss suffered by $\mathbf{u}$ over the worst example,

$$S = S(\mathbf{u}) = \sup_{1\le t\le T} \ell_t(\mathbf{u}), \tag{22}$$

where clearly S depends explicitly in $\mathbf{u}$, which is omitted for simplicity. We now turn to state our first result.

Theorem 5. *Assume $\|\mathbf{x}_t\| \le 1$ for $t = 1 \ldots T$ and $b > 1$. Assume further that $a_t = \frac{1}{1 - \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1} \mathbf{x}_t}$ for $t = 1 \ldots T$. Then*

$$L_T^{\boldsymbol{a}}(\mathbf{u}) \le L_T(\mathbf{u}) + \frac{b}{b-1} S \ln \left|\frac{1}{b}\mathbf{A}_T\right| \ .$$

The proof follows similar steps to Forster [12]. A detailed proof is given in Appendix C.

Proof sketch: We decompose the weighted loss,

$$L_T^{\boldsymbol{a}}(\mathbf{u}) = L_T(\mathbf{u}) + \sum_t (a_t - 1)\ell_t(\mathbf{u}) \leq L_T(\mathbf{u}) + S\sum_t (a_t - 1) \ . \tag{23}$$

From the definition of a_t we have, $a_t - 1 = a_t^2 \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t \leq \frac{b}{b-1} a_t \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t$ (see (C.1)). Finally, following similar steps to Forster [12] we have, $\sum_{t=1}^T a_t \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t \leq \ln\left|\frac{1}{b}\mathbf{A}_T\right|$ (see (C.2)). ■

Next we show a bound that may be sub-logarithmic if the comparison vector $\mathbf{u}$ suffers sub-linear amount of loss. Such a bound was previously proposed by Orabona et.al [21]. We defer the discussion about the bound after providing the proof below.

Theorem 6. *Assume $\|\mathbf{x}_t\| \leq 1$ for $t = 1 \ldots T$, and $b > 1$. Assume further that*

$$a_t = \frac{1}{1 - \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t} \tag{24}$$

for $t = 1 \ldots T$. Then,

$$L_T^{\boldsymbol{a}}(\mathbf{u}) \leq L_T(\mathbf{u}) + \frac{b}{b-1} Sd\left[1 + \ln\left(1 + \frac{L_T(\mathbf{u})}{Sd}\right)\right] \ . \tag{25}$$

We prove the theorem with a refined bound on the sum $\sum_t (a_t - 1)\ell_t(\mathbf{u})$ of (23) using the following two lemmas. In Theorem 5 we bound the loss of all examples with S and then bound the remaining term. Here, instead we show a relation to a subsequence "pretending" all examples of it as suffering a loss S, yet with the same cumulative loss, yielding an effective shorter sequence, which we then bound. In the next lemma we show how to find this subsequence, and in the following one bound the performance.

Lemma 7. *Let $I \subset \{1 \ldots T\}$ be the indices of the $T^{'} = \left\lceil \sum_{t=1}^T \ell_t(\mathbf{u})/S \right\rceil$ largest elements of a_t, that is $|I| = T'$ and $\min_{t \in I} a_t \geq a_\tau$ for all $\tau \in \{1 \ldots T\}/I$. Then,*

$$\sum_{t=1}^T \ell_t(\mathbf{u})(a_t - 1) \leq S\sum_{t \in I}(a_t - 1) \ .$$

Proof: For a vector $\boldsymbol{v} \in \mathbb{R}^T$ define by $I(\boldsymbol{v})$ the set of indicies of the $T^{'}$ maximal absolute-valued elements of $\boldsymbol{v}$, and define $f(\boldsymbol{v}) = \sum_{t \in I(\boldsymbol{v})} |\boldsymbol{v}_t|$. The function $f(\boldsymbol{v})$ is a norm [10] with a dual norm $g(\boldsymbol{h}) = \max\left\{\|\boldsymbol{h}\|_\infty, \frac{\|\boldsymbol{h}\|_1}{T'}\right\}$. From the property of dual norms we have $\boldsymbol{v} \cdot \boldsymbol{h} \leq f(\boldsymbol{v})g(\boldsymbol{h})$. Applying this inequality to $\boldsymbol{v} = (a_1 - 1, \ldots, a_T - 1)$ and $\boldsymbol{h} = (\ell_1(\mathbf{u}), \ldots, \ell_T(\mathbf{u}))$ we get,

$$\sum_{t=1}^T \ell_t(\mathbf{u})(a_t - 1) \leq \max\left\{S, \frac{\sum_{t=1}^T \ell_t(\mathbf{u})}{T'}\right\}\sum_{t \in I}(a_t - 1) \ .$$

Combining with $ST' = S\left\lceil \sum_{t=1}^T \ell_t(\mathbf{u})/S \right\rceil \geq \sum_{t=1}^T \ell_t(\mathbf{u})$, completes the proof. ■

Note that the quantity $\sum_{t\in I} a_t$ is based only on T' examples, yet was generated using all T examples. In fact by running the algorithm with only these T' examples the corresponding sum cannot get smaller. Specifically, assume the algorithm is run with inputs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ and generated a corresponding sequence $(a_1, \ldots, a_T)$. Let I be the set of indices with maximal values of a_t as before. Assume the algorithm is run with the subsequence of examples from I (with the same order) and generated $\alpha_1, \ldots, \alpha_T$ (where we set $\alpha_t = 0$ for $t \notin I$). Then, $\alpha_t \geq a_t$ for all $t \in I$. This statement follows from (3) from which we get that the matrix $\mathbf{A}_t$ is monotonically increasing in t. Thus, by removing examples we get another smaller matrix which leads to a larger value of α_t.

We continue the analysis with a sequence of length T' rather than a subsequence of the original sequence of length T being analyzed. The next lemma upper bounds the sum $\sum_t^{T'} a_t$ over T' inputs with another sum of same length, yet using orthonormal set of vectors of size d.

Lemma 8. *Let $\mathbf{x}_1, \ldots, \mathbf{x}_\tau$ be any τ inputs with unit-norm. Assume the algorithm is performing updates using* (24) *for some $\mathbf{A}_0$ resulting in a sequence $a_1, \ldots, a_\tau$. Let $E = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d\} \subset \mathbb{R}^d$ be an eigen-decomposition of $\mathbf{A}_0$ with corresponding eigenvalues $\lambda_1, \ldots, \lambda_d$. Then there exists a sequence of indices $j_1, \ldots, j_\tau$, where $j_i \in \{1, \ldots, d\}$, such that $\sum_t a_t \leq \sum_t \alpha_t$, where α_t are generated using* (24) *on the sequence $\boldsymbol{v}_{j_1}, \ldots, \boldsymbol{v}_{j_\tau}$.*

Additionally, let n_s be the number of times eigenvector $\boldsymbol{v}_s$ is used ($s = 1, \ldots, d$), that is $n_s = |\{j_t \,:\, j_t = s\}|$ (and $\sum_s n_s = \tau$), then,

$$\sum_t \alpha_t \leq \tau + \sum_{s=1}^{d} \sum_{r=1}^{n_s} \frac{1}{\lambda_s + r - 2} \ .$$

Proof: By induction over τ. For $\tau = 1$ we want to upper bound $a_1 = 1/(1 - \mathbf{x}_1^\top \mathbf{A}_0^{-1} \mathbf{x}_1)$ which is maximized when $\mathbf{x}_1 = \boldsymbol{v}_d$ the eigenvector with minimal eigenvalue λ_d, in this case we have $\alpha_1 = 1/(1 - 1/\lambda_d) = 1 + 1/(\lambda_d - 1)$, as desired.

Next we assume the lemma holds for some $\tau - 1$ and show it for τ. Let $\mathbf{x}_1$ be the first input, and let $\{\gamma_s\}$ and $\{\mathbf{u}_s\}$ be the eigen-values and eigen-vectors of $\mathbf{A}_1 = \mathbf{A}_0 + a_1 \mathbf{x}_1 \mathbf{x}_1^\top$. The assumption of induction implies that $\sum_{t=2}^{\tau} \alpha_t \leq (\tau - 1) + \sum_{s=1}^{d} \sum_{r=1}^{n_s} \frac{1}{\gamma_s + r - 2}$. From Theorem 8.1.8 of [14] we know that the eigenvalues of $\mathbf{A}_1$ satisfy $\gamma_s = \lambda_s + m_s$ for some $m_s \geq 0$ and $\sum_s m_s = 1$. We thus conclude that

$$\sum_t a_t \leq 1 + 1/(\lambda_d - 1) + (\tau - 1) + \sum_{s=1}^{d} \sum_{r=1}^{n_s} \frac{1}{\lambda_s + m_s + r - 2} \ .$$

The last term is convex in $m_1, \ldots, m_d$ and thus is maximized over a vertex of the simplex, that is when $m_k = 1$ for some k and zero otherwise. In this case, the eigen-vectors $\{\mathbf{u}_s\}$ of $\mathbf{A}_1$ are in fact the eigenvectors $\{\boldsymbol{v}_s\}$ of $\mathbf{A}_0$, and the proof is completed. ■

Equipped with these lemmas we now prove Theorem 6.

Algorithm	Bound on Regret $R_T(\mathbf{u})$
Vovk [27]	$b\left\|\mathbf{u}\right\|^2 + dY^2 \ln\left(1+\frac{T}{db}\right)$
Forster [12]	$b\left\|\mathbf{u}\right\|^2 + dY^2 \ln\left(1+\frac{T}{db}\right)$
Crammer et.al. [9]	$rb\left\|\mathbf{u}\right\|^2 + dA \ln\left(1+\frac{T}{drb}\right)$
Orabona et.al. [21]	$2\left\|\mathbf{u}\right\|^2 + d(U+Y)^2 \ln\left(1+\frac{2\left\|\mathbf{u}\right\|^2+\sum_t \ell_t(\mathbf{u})}{d(U+Y)^2}\right)$
Theorem 5	$b\left\|\mathbf{u}\right\|^2 + Sd\frac{b}{b-1} \ln\left(1+\frac{T}{d(b-1)}\right)$
Theorem 6	$b\left\|\mathbf{u}\right\|^2 + Sd\frac{b}{b-1} \ln\left(1+\frac{L_T(\mathbf{u})}{Sd}\right)$

Table 2: Comparison of regret bounds for online regression

Proof: Let $T^{'} = \left\lceil \sum_{t=1}^T \ell_t / S \right\rceil$. Our starting point is the equality $L_T^{\boldsymbol{a}}(\mathbf{u}) = L_T(\mathbf{u}) + \sum_{t=1}^T \ell_t\left(\mathbf{u}\right)\left(a_t - 1\right)$ stated in (23). From Lemma 7 we get,

$$\sum_{t=1}^T \ell_t\left(\mathbf{u}\right)\left(a_t-1\right) \le S\sum_{t\in I}\left(a_t-1\right) \le S\sum_{t}^{T'}\left(\alpha_t-1\right) \ , \tag{26}$$

where I is the subset of T' indices for which a_t are maximal, and α_t are the resulting coefficients computed with (24) using only the sub-sequence of examples $\mathbf{x}_t$ with $t \in I$.

By definition $\mathbf{A}_0 = b\mathbf{I}$ and thus from Lemma 8 we further bound (26) with,

$$\sum_{t=1}^T \ell_t\left(\mathbf{u}\right)\left(a_t-1\right) \le S\sum_{s=1}^d\sum_{r=1}^{n_s}\frac{1}{b+r-2} \ , \tag{27}$$

for some n_s such that $\sum_s n_s = T'$. The last equation is maximized when all the counts n_s are about (as d may not divide T') the same, and thus we further bound (27) with,

$$\begin{aligned}\sum_{t=1}^T \ell_t\left(\mathbf{u}\right)\left(a_t-1\right) \le \ & S\sum_{s=1}^d\sum_{r=1}^{\lceil T'/d\rceil}\frac{1}{b+r-2} \le Sd\sum_{r=1}^{\lceil T'/d\rceil}\frac{b}{b-1}\frac{1}{r}\\ \le \ & Sd\frac{b}{b-1}\left(1+\ln\left(\left\lceil\frac{T'}{d}\right\rceil\right)\right)\\ \le \ & Sd\frac{b}{b-1}\left(1+\ln\left(1+\frac{L_T\left(\mathbf{u}\right)}{Sd}\right)\right) \ ,\end{aligned}$$

which completes the proof. ■

It is instructive to compare bounds of similar algorithms, summarized in Table 2. Our first bound[2] of Theorem 5 is most similar to the bounds of Forster [12], Vovk [27] and Crammer et.al. [9]. Forster and Vovk have a multiplicative factor Y^2 of the logarithm, Crammer et.al. have the factor $A = \sup_{1\le t\le T}\ell_t(\text{alg})$, and

[2] The bound in the table is obtained by noting that log det is a concave function of the eigenvalues of the matrix, upper bounded when all the eigenvalues are equal (with the same trace).

we have the worst-loss of $\mathbf{u}$ over all examples (denoted by S). Thus, our first bound is better than the bound of Crammer et.al. (as often $S < A$), and better than the bounds of Forster and Vovk on problems that are approximately linear $y_t \approx \mathbf{u} \cdot \mathbf{x}_t$ for $t = 1, \ldots, T$ and Y is large, while their bound is better if Y is small. Note that the analysis of Forster [12] assumes that the labels y_t are bounded, and formally the algorithm should know this bound, while Crammer et.al. assume that the inputs are bounded, as we do.

Our second bound of Theorem 6 is similar to the bound of Orabona et.al. [21]. Both bounds have potentially sub-logarithmic regret as the cumulative loss $L(\mathbf{u})$ may be sublinear in T. Yet, their bound has a multiplicative factor of $(U + Y)^2$, while our bound has only the maximal loss S, which, as before, can be much smaller. Additionally, their analysis assumes that both the inputs $\mathbf{x}_t$ and the labels y_t are bounded, while we only assume that the inputs are bounded, and furthermore, our algorithm does not need to assume and know a compact set which contains $\mathbf{u}$ ($\|\mathbf{u}\| \leq U$), as opposed to their algorithm.

5. Learning in Non-Stationary Environment

In this section we present a generalization of the last-step min-max predictor for non-stationary problems given in (2). We define the predictor to be,

$$\hat{y}_T = \arg\min_{\hat{y}_T} \max_{y_T} \left[\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 - \inf_{\mathbf{u}_1,\ldots,\mathbf{u}_T,\bar{\mathbf{u}}} \left(b \|\bar{\mathbf{u}}\|^2 + cV_m + L_T^{\tilde{\mathbf{a}}}(\mathbf{u}_1, \ldots, \mathbf{u}_T) \right) \right] \tag{28}$$

for

$$V_m = \sum_{t=1}^{T} \|\mathbf{u}_t - \bar{\mathbf{u}}\|^2 \ , \tag{29}$$

positive constants $b, c > 0$ and weights $\tilde{a}_t \geq 1$ for $1 \leq t \leq T$.

As mentioned above, we use an extended notion of function class, using different vectors $\mathbf{u}_t$ across time T. We circumvent here the problem mentioned in the end of Sec. 2, and restrict the adversary from choosing an arbitrary T-tuple $(\mathbf{u}_1, \ldots, \mathbf{u}_T)$ by introducing a reference weight-vector $\bar{\mathbf{u}}$. Specifically, indeed we replace the single-weight cumulative-loss $L_T^{\mathbf{a}}(\mathbf{u})$ in (2) with a multi-weight cumulative-loss $L_T^{\tilde{\mathbf{a}}}(\mathbf{u}_1, \ldots, \mathbf{u}_T)$ in (28), yet, we add the term cV_m to (28) penalizing a T-tuple $(\mathbf{u}_1, \ldots, \mathbf{u}_T)$ that its elements $\{\mathbf{u}_t\}$ are far from some single point $\bar{\mathbf{u}}$. Intuitively, V_m serves as a measure of complexity of the T-tuple by measuring the deviation of its elements from some vector.

The new formulation of (28) clearly subsumes the formulation of (2), as if $\mathbf{u}_1 = \ldots \mathbf{u}_T = \bar{\mathbf{u}} = \mathbf{u}$, then (28) reduces to (2). We now show that in-fact the two notions of last-step min-max predictors are equivalent. The following lemma characterizes the solution of the inner infimum of (28) over $\bar{\mathbf{u}}$.

Lemma 9. *For any $\bar{\mathbf{u}} \in \mathbb{R}^d$, the function*

$$J(\mathbf{u}_1, \ldots, \mathbf{u}_T) = b \|\bar{\mathbf{u}}\|^2 + c \sum_{t=1}^{T} \|\mathbf{u}_t - \bar{\mathbf{u}}\|^2 + \sum_{t=1}^{T} \tilde{a}_t \left(y_t - \mathbf{u}_t^\top \mathbf{x}_t \right)^2 \ ,$$

is minimal for

$$\mathbf{u}_t = \bar{\mathbf{u}} + \frac{c^{-1}}{\widetilde{a}_t^{-1} + c^{-1}\left\|x_t\right\|^2}\left(y_t - \bar{\mathbf{u}}^\top \mathbf{x}_t\right)\mathbf{x}_t$$

for $t = 1...T$*. The minimal value of* $J\left(\mathbf{u}_1, \ldots, \mathbf{u}_T\right)$ *is given by*

$$J_{min} = b\left\|\bar{\mathbf{u}}\right\|^2 + \sum_{t=1}^{T} \frac{1}{\widetilde{a}_t^{-1} + c^{-1}\left\|\mathbf{x}_t\right\|^2}\left(y_t - \bar{\mathbf{u}}^\top \mathbf{x}_t\right)^2 \ . \tag{30}$$

The proof appears in Appendix D.

Remark 2. *The minimization problem in Lemma 9 can be interpreted as MAP estimator of* $\bar{\mathbf{u}}$ *based on the sequence* $\left\{(\mathbf{x}_t, y_t)\right\}_{t=1}^{T}$ *in the following generative model:*

$$\begin{aligned}
\bar{\mathbf{u}} &\sim N\left(0, \sigma_b^2 \mathbf{I}\right) \\
\mathbf{u}_t &\sim N\left(\bar{\mathbf{u}}, \sigma_c^2 \mathbf{I}\right) \\
y_t &\sim N\left(\mathbf{x}_t^\top \mathbf{u}_t, \sigma_t^2\right) \ ,
\end{aligned}$$

where $\sigma_b^2 = \frac{1}{2b}$*,* $\sigma_c^2 = \frac{1}{2c}$ *and* $\sigma_t^2 = \frac{1}{2\widetilde{a}_t}$*.*

Indeed,

$$\begin{aligned}
\bar{\mathbf{u}}_{MAP} &= \arg\max_{\bar{\mathbf{u}}} P\left(\bar{\mathbf{u}} \mid \left\{\mathbf{u}_t\right\}, \left\{\mathbf{x}_t\right\}, \left\{y_t\right\}\right) \\
&= \arg\max_{\bar{\mathbf{u}}} \left[P\left(\bar{\mathbf{u}}\right) \prod_{t=1}^{T} P\left(\mathbf{u}_t \mid \bar{\mathbf{u}}\right) \prod_{t=1}^{T} P\left(y_t \mid \mathbf{u}_t, \mathbf{x}_t\right)\right] \\
&= \arg\min_{\bar{\mathbf{u}}} \left[-\log P\left(\bar{\mathbf{u}}\right) - \sum_{t=1}^{T} \log P\left(\mathbf{u}_t \mid \bar{\mathbf{u}}\right) - \sum_{t=1}^{T} \log P\left(y_t \mid \mathbf{u}_t, \mathbf{x}_t\right)\right] \ .
\end{aligned} \tag{31}$$

By our gaussian generative model we have

$$\begin{aligned}
-\log P\left(\bar{\mathbf{u}}\right) &= \log\left(2\pi\sigma_b^2\right)^{d/2} + \frac{1}{2\sigma_b^2}\left\|\bar{\mathbf{u}}\right\|^2 \\
-\log P\left(\mathbf{u}_t \mid \bar{\mathbf{u}}\right) &= \log\left(2\pi\sigma_c^2\right)^{d/2} + \frac{1}{2\sigma_c^2}\left\|\mathbf{u}_t - \bar{\mathbf{u}}\right\|^2 \\
-\log P\left(y_t \mid \mathbf{u}_t, \mathbf{x}_t\right) &= \log\left(2\pi\sigma_t^2\right)^{1/2} + \frac{1}{2\sigma_t^2}\left(y_t - \mathbf{x}_t^\top \mathbf{u}_t\right)^2 \ .
\end{aligned}$$

Substituting in (31) *we get*

$$\bar{\mathbf{u}}_{MAP} = \arg\min_{\bar{\mathbf{u}}} \left[\frac{1}{2\sigma_b^2}\left\|\bar{\mathbf{u}}\right\|^2 + \frac{1}{2\sigma_c^2}\sum_{t=1}^{T}\left\|\mathbf{u}_t - \bar{\mathbf{u}}\right\|^2 + \sum_{t=1}^{T}\frac{1}{2\sigma_t^2}\left(y_t - \mathbf{x}_t^\top \mathbf{u}_t\right)^2\right] \ ,$$

and by using $\frac{1}{2\sigma_b^2} = b$*,* $\frac{1}{2\sigma_c^2} = c$*,* $\frac{1}{2\sigma_t^2} = \widetilde{a}_t$ *we get the minimization problem in Lemma 9.*

Substituting (30) in (28) we obtain the following form of the last-step minmax predictor,

$$\hat{y}_T = \arg\min_{\hat{y}_T}\max_{y_T}\left[\sum_{t=1}^{T}(y_t - \hat{y}_t)^2 - \inf_{\bar{\mathbf{u}} \in \mathbb{R}^d}\left(b\left\|\bar{\mathbf{u}}\right\|^2 + \sum_{t=1}^{T}\frac{1}{\widetilde{a}_t^{-1} + c^{-1}\left\|\mathbf{x}_t\right\|^2}\left(y_t - \mathbf{x}_t^\top \bar{\mathbf{u}}\right)^2\right)\right] \ . \tag{32}$$

Clearly, both equations (32) and (2) are equivalent when identifying,

$$a_t = \frac{1}{\widetilde{a}_t^{-1} + c^{-1}\left\|\mathbf{x}_t\right\|^2} \ . \tag{33}$$

Therefore, we can use the results of the previous sections.

Corollary 10. *The optimal prediction for the last round T is $\hat{y}_T = \mathbf{b}_{T-1}^\top \mathbf{A}_{T-1}^{-1}\mathbf{x}_T$ if the following condition is hold*

$$1 + a_T \mathbf{x}_T^\top \mathbf{A}_{T-1}^{-1}\mathbf{x}_T - a_T \leq 0 \ ,$$

where a_T defined by (33) *and where we replace* (3) *with*

$$\mathbf{A}_t = b\mathbf{I} + \sum_{s=1}^{t} a_s \mathbf{x}_s \mathbf{x}_s^\top = b\mathbf{I} + \sum_{s=1}^{t} \frac{1}{\widetilde{a}_s^{-1} + c^{-1}\left\|\mathbf{x}_s\right\|^2}\mathbf{x}_s\mathbf{x}_s^\top$$

and (4) *with,*

$$\mathbf{b}_t = \sum_{s=1}^{t} a_s y_s \mathbf{x}_s = \sum_{s=1}^{t} \frac{1}{\widetilde{a}_s^{-1} + c^{-1}\left\|\mathbf{x}_s\right\|^2} y_s \mathbf{x}_s.$$

Although most of the analysis above holds for $1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t - a_t \leq 0$ in the end of the day, Theorem 5 assumed that this inequality holds as equality. Substituting $a_t = \frac{1}{1 - \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t}$ in (33) and solving for $\widetilde{a}_t$ we obtain,

$$\widetilde{a}_t = \frac{1}{1 - \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t - c^{-1}\left\|\mathbf{x}_t\right\|^2} \ . \tag{34}$$

The last-step minmax predictor (28) is convex if $\widetilde{a}_t \geq 0$, which holds if, $1/b + 1/c \leq 1$,because $\mathbf{A}_{t-1}^{-1} \preceq \mathbf{A}_0^{-1} = (1/b)\mathbf{I}$ and we assume that $\left\|\mathbf{x}_t\right\|^2 \leq 1$.

Let us state the analogous statements of Theorem 4 and Theorem 5. Substituting Lemma 9 in Theorem 4 we bound the cumulative loss of the algorithm with the weighted loss of any T-tuple $(\mathbf{u}_1, \ldots, \mathbf{u}_T)$.

Corollary 11. *Assume $\|\mathbf{x}_t\| \leq 1$, $1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t - a_t \leq 0$ for $t = 1 \ldots T$, and $1/b + 1/c \leq 1$. Then, the loss of the last-step minmax predictor, $\hat{y}_t = \mathbf{b}_{t-1}^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t$ for $t = 1 \ldots T$, is upper bounded by,*

$$L_T(\mathtt{WEMM}) \leq \inf_{\mathbf{u}\in\mathbb{R}^d}\left(b\left\|\mathbf{u}\right\|^2 + L_T^{\mathbf{a}}(\mathbf{u})\right) = \inf_{\mathbf{u}_1,\ldots,\mathbf{u}_T,\bar{\mathbf{u}}}\left(b\left\|\bar{\mathbf{u}}\right\|^2 + c\sum_{t=1}^{T}\left\|\mathbf{u}_t - \bar{\mathbf{u}}\right\|^2 + L_T^{\widetilde{\mathbf{a}}}(\mathbf{u}_1, \ldots, \mathbf{u}_T)\right) \ .$$

Furthermore, if $1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t - a_t = 0$, then the last inequality is in fact an equality.

Next we relate the weighted cumulative loss $L_T^{\widetilde{\mathbf{a}}}(\mathbf{u}_1, \ldots, \mathbf{u}_T)$ to the loss itself $L_T(\mathbf{u}_1, \ldots, \mathbf{u}_T)$,

Corollary 12. *Assume $\|\mathbf{x}_t\| \leq 1$ for $t = 1 \ldots T$, $b > 1$ and $1/b + 1/c \leq 1$. Assume additionally that $\widetilde{a}_t = \frac{1}{1 - \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t - c^{-1}\|\mathbf{x}_t\|^2}$ as given in* (34)*. Then*

$$L_T^{\widetilde{\mathbf{a}}}(\mathbf{u}_1, \ldots, \mathbf{u}_T) \leq L_T(\mathbf{u}_1, \ldots, \mathbf{u}_T) + \frac{b}{b-1} S \ln\left|\frac{1}{b}\mathbf{A}_T\right| + TS\frac{1}{c\left(1 - b^{-1}\right)^2 - \left(1 - b^{-1}\right)} \ .$$

Proof: We start as in the proof of Theorem 5 and decompose the weighted loss,

$$\begin{aligned} L_T^{\tilde{\mathbf{a}}}(\mathbf{u}_1,\ldots,\mathbf{u}_T) &= L_T(\mathbf{u}_1,\ldots,\mathbf{u}_T) + \sum_t (\tilde{a}_t - 1)\ell_t(\mathbf{u}_t) \\ &\le L_T(\mathbf{u}_1,\ldots,\mathbf{u}_T) + S\sum_t (a_t - 1) + S\sum_t (\tilde{a}_t - a_t) \ . \end{aligned} \tag{35}$$

We bound the sum of the third term,

$$\begin{aligned} \tilde{a}_t - a_t &= \frac{1}{1 - \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t - c^{-1}\left\|\mathbf{x}_t\right\|^2} - \frac{1}{1 - \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t} \\ &= \frac{c^{-1}\left\|\mathbf{x}_t\right\|^2}{\left(1 - \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t - c^{-1}\left\|\mathbf{x}_t\right\|^2\right)\left(1 - \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t\right)} \\ &\le \frac{c^{-1}}{\left(1 - b^{-1} - c^{-1}\right)\left(1 - b^{-1}\right)} = \frac{1}{c\left(1 - b^{-1}\right)^2 - \left(1 - b^{-1}\right)} \ . \end{aligned} \tag{36}$$

Additionally, as in Theorem 5 the second term is bounded with $\frac{b}{b-1} S \ln\left|\frac{1}{b}\mathbf{A}_T\right|$. Substituting this bound and (36) in (35) completes the proof. ■

Combining the last two corollaries yields the main result of this section.

Corollary 13. *Under the conditions of Corollary 12 the cumulative loss of the last-step minmax predictor is upper bounded by,*

$$L_T(\mathit{WEMM}) \le \inf_{\mathbf{u}_1,\ldots,\mathbf{u}_T,\bar{\mathbf{u}}} \left(b\left\|\bar{\mathbf{u}}\right\|^2 + cV_m + L_T(\mathbf{u}_1,\ldots,\mathbf{u}_T) + \frac{Sb}{b-1}\ln\left|\frac{1}{b}\mathbf{A}_T\right| + \frac{TS}{c\left(1-b^{-1}\right)^2 - \left(1-b^{-1}\right)} \right) \ ,$$

where V_m is the deviation of $\{\mathbf{u}_t\}$ from some fixed weight-vector as defined in (29)*. Additionally, setting $c_V = \frac{b}{b-1}\left(1 + \sqrt{\frac{ST}{V_m}}\right)$ minimizing the above bound over c,*

$$L_T(\mathit{WEMM}) \le \inf_{\mathbf{u}_1,\ldots,\mathbf{u}_T,\bar{\mathbf{u}}} \left(b\left\|\bar{\mathbf{u}}\right\|^2 + L_T(\mathbf{u}_1,\ldots,\mathbf{u}_T) + \frac{Sb}{b-1}\ln\left|\frac{1}{b}\mathbf{A}_T\right| + \frac{b}{b-1}\left(V_m + 2\sqrt{STV_m}\right) \right) \ .$$

Few comments. First, it is straightforward to verify that $c_V = \frac{b}{b-1}\left(1 + \sqrt{\frac{ST}{V_m}}\right)$ satisfy the constraint $1/b + 1/c_V \le 1$. Second, this bound strictly generalizes the bound for the stationary case, since Corollary 12 reduces to Theorem 5 when all the weight-vectors equal each other $\mathbf{u}_1 = \ldots \mathbf{u}_T = \bar{\mathbf{u}}$ (i.e. $V_m = 0$). Third, the constant c (or c_V) is not used by the algorithm, but only in the analysis. So there is no need to know the actual deviation V_m to tune the algorithm. In other words, the bound applies essentially to the same last step minmax predictor defined in Theorem 2. Finally, we have a bound for the non-stationary case based on Theorem 6 instead of Theorem 5, by replacing the term

$$\frac{Sb}{b-1}\ln\left|\frac{1}{b}\mathbf{A}_T\right| \ ,$$

with

$$\frac{Sbd}{b-1}\left(1 + \ln\left(1 + \frac{\sum_t \ell_t(\mathbf{u}_t)}{Sd}\right)\right) \ .$$

6. Related work

The problem of predicting reals in an online manner was studied for more than five decades. Clearly we cannot cover all previous work here, and the reader is refered to the encyclopedic book of Cesa-Bianchi and Lugosi [7] for a full survey.

Widrow and Hoff [28] studied a gradient descent algorithm for the squared loss. Many variants of the algorithm were studied since then. A notable example is the normalized least mean squares algorithm (NLMS) [2, 3] that adapts to the input's scale. More gradient descent based algorithms and bounds for regression with the squared loss were proposed by Cesa-Bianchi et.al. [5] about two decades ago. These algorithms were generalized and extended by Kivinen and Warmuth [19] using additional regularization functions.

An online version of the ridge regression algorithm in the worst-case setting was proposed and analyzed by Foster [13]. A related algorithm called the Aggregating Algorithm (AA) was studied by Vovk [26]. See also the work of Azoury and Warmuth [1].

The recursive least squares (RLS) [15] is a similar algorithm proposed for adaptive filtering. A variant of the RLS algorithm (AROW for regression [25]) was analysed by Crammer et.al. [9]. All algorithms make use of second order information, as they maintain a weight-vector and a covariance-like positive semi-definite (PSD) matrix used to re-weight the input. The eigenvalues of this covariance-like matrix grow with time t, a property which is used to prove logarithmic regret bounds. Orabona et.al. [21] showed that beyond logarithmic regret bound can be achieved when the total best linear model loss is sublinear in T. We derive a similar bound, with a multiplicative factor that depends on the worst-loss of $\mathbf{u}$, rather than a bound Y on the labels. Hazan and Kale [16] developed regret bounds that depend logarithmically on the variance of the side information used to define the loss sequence. In the regression case, this corresponds to a bound that depends on the variance of the instance vectors $\mathbf{x}_t$, rather than on the loss of the competitor, as the bound of Orabona et.al. [21] and our bound.

The derivation of our algorithm shares similarities with the work of Forster [12]. Both algorithms are motivated from the last-step min-max predictor. Yet, the formulation of Forster [12] yields a convex optimization for which the max operation over y_t is not bounded, and thus he used an artificial clipping operation to avoid unbounded solutions. With a proper tuning of a_t and a weighted loss, we are able to obtain a problem that is convex in $\hat{y}_t$ and concave in y_t, and thus well defined.

Most recent work is focused in the stationary setting. We also discuss a specific weak-notion of non-stationary setting, for which the few weight-vectors can be used for comparison and their total deviation is computed with respect to some single weight-vector. Recently, Vaits and Crammer [25] proposed an algorithm designed for non-stationary environments. Herbster and Warmuth [17] discussed general gradient descent algorithms with projection of the weight-vector using the Bregman divergence, and Zinkevich [29] developed an algorithm for online convex programming. Busuttil and Kalnishkan [4] developed a variant of the aggregating algorithm in the non-stationary environment. They all use a stronger notion of diversity between vectors, as their distance is measured with consecutive vectors (that is drift that may end far from the starting point). Thus, the bounds in these papers cannot be compared in general to our bound in Corollary 13. The H_∞ filters (see e.g. papers by Simon [22, 23]) are a family of (robust) linear filters developed based on a min-max approach, like WEMM, and analyzed in the worst case setting. These filters are reminiscent of the celebrated Kalman filter [18], which was motivated and analyzed in a stochastic setting with Gaussian noise. Finally, few second-order algorithms were recently proposed in other contexts [6, 8, 11, 20].

7. Summary and Conclusions

We proposed a modification of the last-step min-max algorithm [12] using weights over examples, and showed how to choose these weights for the problem to be well defined – convex – which enabled us to develop the last step min-max predictor, without requiring the labels to be bounded. Our algorithmic formulations depend on inner- and outer-products and thus can be employed with kernel functions. Our analysis bounds the regret with quantities that depend only on the loss of the competitor, with no need for any knowledge of the problem. Our prediction algorithm was motivated from the last-step minmax predictor problem for stationary setting, but we showed that the same algorithm can be used to derive a bound for a class of *non-stationary* problems as well.

An interesting direction would be to extend the algorithm for general loss functions rather than the squared loss, or to classification tasks.

Acknowledgements

The research is partially supported by an Israeli Science Foundation grant ISF- 1567/10.

Appendix A. Proof of Lemma 3

Proof: Using the Woodbury identity we get

$$\mathbf{A}_t^{-1} = \mathbf{A}_{t-1}^{-1} - \frac{\mathbf{A}_{t-1}^{-1}\mathbf{x}_t\mathbf{x}_t^\top\mathbf{A}_{t-1}^{-1}}{\frac{1}{a_t} + \mathbf{x}_t^\top\mathbf{A}_{t-1}^{-1}\mathbf{x}_t} ,$$

therefore the left side of (9) is

$$\begin{aligned}
a_t^2 \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t + 1 - a_t &= a_t^2 \mathbf{x}_t^\top \left(\mathbf{A}_{t-1}^{-1} - \frac{\mathbf{A}_{t-1}^{-1}\mathbf{x}_t\mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}}{\frac{1}{a_t} + \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t} \right) \mathbf{x}_t + 1 - a_t \\
&= a_t^2 \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t - \frac{a_t^2 \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t\mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t}{\frac{1}{a_t} + \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t} + 1 - a_t \\
&= \frac{1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t - a_t}{1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t} \ .
\end{aligned}$$

■

Appendix B. Proof of Theorem 4

Proof: Using the Woodbury matrix identity we get

$$\mathbf{A}_t^{-1} = \mathbf{A}_{t-1}^{-1} - \frac{\mathbf{A}_{t-1}^{-1}\mathbf{x}_t\mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}}{\frac{1}{a_t} + \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t} \ , \tag{B.1}$$

therefore

$$\mathbf{A}_t^{-1}\mathbf{x}_t = \mathbf{A}_{t-1}^{-1}\mathbf{x}_t - \frac{\mathbf{A}_{t-1}^{-1}\mathbf{x}_t\mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t}{\frac{1}{a_t} + \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t} = \frac{\mathbf{A}_{t-1}^{-1}\mathbf{x}_t}{1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t} \ . \tag{B.2}$$

For $t = 1 \ldots T$ we have

$$\begin{aligned}
& \ell_t(\texttt{WEMM}) + \inf_{\mathbf{u}\in\mathbb{R}^d} \left(b \left\|\mathbf{u}\right\|^2 + L_{t-1}^{\boldsymbol{a}}(\mathbf{u}) \right) - \inf_{\mathbf{u}\in\mathbb{R}^d} \left(b \left\|\mathbf{u}\right\|^2 + L_t^{\boldsymbol{a}}(\mathbf{u}) \right) \\
= \ & (y_t - \hat{y}_t)^2 + \inf_{\mathbf{u}\in\mathbb{R}^d} \left(b\left\|\mathbf{u}\right\|^2 + \sum_{s=1}^{t-1} a_s \left(y_s - \mathbf{u}^\top \mathbf{x}_s\right)^2 \right) - \inf_{\mathbf{u}\in\mathbb{R}^d} \left(b\left\|\mathbf{u}\right\|^2 + \sum_{s=1}^{t} a_s \left(y_s - \mathbf{u}^\top \mathbf{x}_s\right)^2 \right) \\
\overset{(5)}{=} \ & (y_t - \hat{y}_t)^2 + \sum_{s=1}^{t-1} a_s y_s^2 - \mathbf{b}_{t-1}^\top \mathbf{A}_{t-1}^{-1}\mathbf{b}_{t-1} - \sum_{s=1}^{t} a_s y_s^2 + \mathbf{b}_t^\top \mathbf{A}_t^{-1}\mathbf{b}_t \\
= \ & (y_t - \hat{y}_t)^2 - a_t y_t^2 - \mathbf{b}_{t-1}^\top \mathbf{A}_{t-1}^{-1}\mathbf{b}_{t-1} + \mathbf{b}_t^\top \mathbf{A}_t^{-1}\mathbf{b}_t \\
\overset{(10)}{=} \ & (y_t - \hat{y}_t)^2 - a_t y_t^2 - \mathbf{b}_{t-1}^\top \mathbf{A}_{t-1}^{-1}\mathbf{b}_{t-1} + \mathbf{b}_{t-1}^\top \mathbf{A}_t^{-1}\mathbf{b}_{t-1} + 2a_t y_t \mathbf{b}_{t-1}^\top \mathbf{A}_t^{-1}\mathbf{x}_t + a_t^2 y_t^2 \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t \\
= \ & (y_t - \hat{y}_t)^2 - a_t y_t^2 - \mathbf{b}_{t-1}^\top \left(\mathbf{A}_{t-1}^{-1} - \mathbf{A}_t^{-1}\right)\mathbf{b}_{t-1} + 2a_t y_t \mathbf{b}_{t-1}^\top \mathbf{A}_t^{-1}\mathbf{x}_t + a_t^2 y_t^2 \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t \\
\overset{(14)}{=} \ & (y_t - \hat{y}_t)^2 - a_t y_t^2 - \mathbf{b}_{t-1}^\top \mathbf{A}_t^{-1} a_t \mathbf{x}_t\mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{b}_{t-1} + 2a_t y_t \mathbf{b}_{t-1}^\top \mathbf{A}_t^{-1}\mathbf{x}_t + a_t^2 y_t^2 \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t \\
= \ & (y_t - \hat{y}_t)^2 - a_t y_t^2 + a_t \left(-\hat{y}_t \mathbf{b}_{t-1}^\top + 2y_t \mathbf{b}_{t-1}^\top + a_t y_t^2 \mathbf{x}_t^\top\right) \mathbf{A}_t^{-1}\mathbf{x}_t \\
\overset{(\text{B.2})}{=} \ & (y_t - \hat{y}_t)^2 - a_t y_t^2 + a_t \left(-\hat{y}_t \mathbf{b}_{t-1}^\top + 2y_t \mathbf{b}_{t-1}^\top + a_t y_t^2 \mathbf{x}_t^\top\right) \frac{\mathbf{A}_{t-1}^{-1}\mathbf{x}_t}{1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t} \\
= \ & (y_t - \hat{y}_t)^2 + a_t \frac{-y_t^2 - y_t^2 a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t - \hat{y}_t^2 + 2y_t\hat{y}_t + a_t y_t^2 \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t}{1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t} \\
= \ & (y_t - \hat{y}_t)^2 - a_t \frac{(y_t - \hat{y}_t)^2}{1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t} \\
= \ & \frac{1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t - a_t}{1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t} (y_t - \hat{y}_t)^2 \leq 0 \ .
\end{aligned}$$

Summing over $t \in \{1, \ldots, T\}$ and using (1) yields $L_T(\texttt{WEMM}) - \inf_{\mathbf{u}\in\mathbb{R}^d}\left(b\left\|\mathbf{u}\right\|^2 + L_T^{\boldsymbol{a}}(\mathbf{u})\right) \leq 0$. ■

Appendix C. Proof of Theorem 5

Proof: From (B.1) we see that $\mathbf{A}_t^{-1} \prec \mathbf{A}_{t-1}^{-1}$ and because $\mathbf{A}_0 = b\mathbf{I}$ we get

$$\mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t < \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t < \mathbf{x}_t^\top \mathbf{A}_{t-2}^{-1}\mathbf{x}_t < \ldots < \mathbf{x}_t^\top \mathbf{A}_0^{-1}\mathbf{x}_t = \frac{1}{b}\left\|\mathbf{x}_t\right\|^2 \leq \frac{1}{b} ,$$

therefore $1 \leq a_t \leq \frac{1}{1-\frac{1}{b}} = \frac{b}{b-1}$. From (B.2) we have

$$\begin{aligned} \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t &= \frac{\mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t}{1 + a_t \mathbf{x}_t^\top \mathbf{A}_{t-1}^{-1}\mathbf{x}_t} \\ &= \frac{1-\frac{1}{a_t}}{1 + a_t\left(1-\frac{1}{a_t}\right)} = \frac{a_t - 1}{a_t^2} , \end{aligned}$$

so we can bound the term $a_t - 1$ as following

$$a_t - 1 = a_t^2 \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t \leq \frac{b}{b-1} a_t \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t . \tag{C.1}$$

With an argument similar to [12] we have, $a_t\mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t \leq \ln\frac{|\mathbf{A}_t|}{\left|\mathbf{A}_t - a_t\mathbf{x}_t\mathbf{x}_t^\top\right|} = \ln\frac{|\mathbf{A}_t|}{|\mathbf{A}_{t-1}|}$. Summing the last inequality over t and using the initial value $\ln\left|\frac{1}{b}\mathbf{A}_0\right| = 0$ we get

$$\sum_{t=1}^{T} a_t \mathbf{x}_t^\top \mathbf{A}_t^{-1}\mathbf{x}_t \leq \ln\left|\frac{1}{b}\mathbf{A}_T\right| . \tag{C.2}$$

Substituting the last equation in (C.1) we get the logarithmic bound $\sum_{t=1}^{T}(a_t - 1) \leq \frac{b}{b-1}\ln\left|\frac{1}{b}\mathbf{A}_T\right|$, as required. ■

Appendix D. Proof of Lemma 9

Proof: We set the derivative of J with respect to $\mathbf{u}_t$ to zero,

$$\frac{\partial J}{\partial \mathbf{u}_t} = 2c\left(\mathbf{u}_t - \bar{\mathbf{u}}\right) - 2\tilde{a}_t\left(y_t - \mathbf{u}_t^\top\mathbf{x}_t\right)\mathbf{x}_t = 0 ,$$

and solve for $\mathbf{u}_t$:

$$\begin{aligned} \mathbf{u}_t &= \left(c\mathbf{I} + \tilde{a}_t\mathbf{x}_t\mathbf{x}_t^\top\right)^{-1}\left(c\bar{\mathbf{u}} + \tilde{a}_t y_t \mathbf{x}_t\right) \\ &= \left(c^{-1}\mathbf{I} - \frac{c^{-2}}{\tilde{a}_t^{-1} + c^{-1}\left\|\mathbf{x}_t\right\|^2}\mathbf{x}_t\mathbf{x}_t^\top\right)\left(c\bar{\mathbf{u}} + \tilde{a}_t y_t\mathbf{x}_t\right) \\ &= \bar{\mathbf{u}} + \frac{c^{-1}}{\tilde{a}_t^{-1} + c^{-1}\left\|\mathbf{x}_t\right\|^2}\left(y_t - \bar{\mathbf{u}}^\top\mathbf{x}_t\right)\mathbf{x}_t . \end{aligned} \tag{D.1}$$

For the optimal $\mathbf{u}_t$ of (D.1), we compute the following two terms, which are used next,

$$\begin{aligned}\left(y_t - \mathbf{u}_t^\top \mathbf{x}_t\right)^2 &= \left(y_t - \left(\bar{\mathbf{u}} + \frac{c^{-1}}{\widetilde{a}_t^{-1} + c^{-1}\left\|\mathbf{x}_t\right\|^2}\left(y_t - \bar{\mathbf{u}}^\top \mathbf{x}_t\right)\mathbf{x}_t\right)^\top \mathbf{x}_t\right)^2 \\ &= \frac{\widetilde{a}_t^{-2}}{\left(\widetilde{a}_t^{-1} + c^{-1}\left\|\mathbf{x}_t\right\|^2\right)^2}\left(y_t - \bar{\mathbf{u}}^\top \mathbf{x}_t\right)^2 \qquad \text{(D.2)} \\ \left\|\mathbf{u}_t - \bar{\mathbf{u}}\right\|^2 &= \left\|\frac{c^{-1}}{\widetilde{a}_t^{-1} + c^{-1}\left\|\mathbf{x}_t\right\|^2}\left(y_t - \bar{\mathbf{u}}^\top \mathbf{x}_t\right)\mathbf{x}_t\right\|^2 \\ &= \frac{c^{-2}}{\left(\widetilde{a}_t^{-1} + c^{-1}\left\|\mathbf{x}_t\right\|^2\right)^2}\left(y_t - \bar{\mathbf{u}}^\top \mathbf{x}_t\right)^2\left\|\mathbf{x}_t\right\|^2 \ . \qquad \text{(D.3)}\end{aligned}$$

From (D.2) and (D.3) we get

$$c\left\|\mathbf{u}_t - \bar{\mathbf{u}}\right\|^2 + \widetilde{a}_t\left(y_t - \mathbf{u}_t^\top \mathbf{x}_t\right)^2 = \frac{1}{\widetilde{a}_t^{-1} + c^{-1}\left\|\mathbf{x}_t\right\|^2}\left(y_t - \bar{\mathbf{u}}^\top \mathbf{x}_t\right)^2 \ .$$

Therefore the minimal value of $J\left(\mathbf{u}_1, \ldots, \mathbf{u}_T\right)$ is given by,

$$J_{min} \quad = \quad b\left\|\bar{\mathbf{u}}\right\|^2 + \sum_{t=1}^{T}\frac{1}{\widetilde{a}_t^{-1} + c^{-1}\left\|\mathbf{x}_t\right\|^2}\left(y_t - \bar{\mathbf{u}}^\top \mathbf{x}_t\right)^2 \ ,$$

which completes the proof. ■

References

[1] Katy S. Azoury and Manfred K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

[2] Neil J. Bershad. Analysis of the normalized lms algorithm with gaussian inputs. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):793–806, 1986.

[3] Robert R. Bitmead and Brian D. O. Anderson. Performance of adaptive estimation algorithms in dependent random environments. *IEEE Transactions on Automatic Control*, 25:788–794, 1980.

[4] Steven Busuttil and Yuri Kalnishkan. Online regression competitive with changing predictors. In *ALT*, pages 181–195, 2007.

[5] Nicolo Ceas-Bianchi, Philip M. Long, and Manfred K. Warmuth. Worst case quadratic loss bounds for on-line prediction of linear functions by gradient descent. Technical Report IR-418, University of California, Santa Cruz, CA, USA, 1993.

[6] Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. A second-order perceptron algorithm. *Siam Journal of Commutation*, 34(3):640–668, 2005.

[7] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, New York, NY, USA, 2006.

[8] Koby Crammer, Alex Kulesza, and Mark Dredze. Adaptive regularization of weighted vectors. In *Advances in Neural Information Processing Systems 23*, 2009.

[9] Koby Crammer, Alex Kulesza, and Mark Dredze. New $\mathcal{H}\infty$ bounds for the recursive least squares algorithm exploiting input structure. In *ICASSP*, pages 2017–2020, 2012.

[10] Ofer Dekel, Philip M. Long, and Yoram Singer. Online learning of multiple tasks with a shared loss. *Journal of Machine Learning Research*, 8:2233–2264, 2007.

[11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *COLT*, pages 257–269, 2010.

[12] Jurgen Forster. On relative loss bounds in generalized linear regression. In *FCT*, 1999.

[13] Dean P. Foster. Prediction in the worst case. *The An. of Stat.*, 19(2):1084–1090, 1991.

[14] Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[15] Monson H. Hayes. 9.4: Recursive least squares. In *Statistical Digital Signal Processing and Modeling*, page 541, 1996.

[16] Elad Hazan and Satyen Kale. On stochastic and worst-case models for investing. In *NIPS*, pages 709–717, 2009.

[17] Mark Herbster and Manfred K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.

[18] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[19] Jyrki Kivinen and Manfred K. Warmuth. Exponential gradient versus gradient descent for linear predictors. *Information and Computation*, 132:132–163, 1997.

[20] Hugh Brendan McMahan and Matthew J. Streeter. Adaptive bound optimization for online convex optimization. In *COLT*, pages 244–256, 2010.

[21] Francesco Orabona, Nicolo Cesa-Bianchi, and Claudio Gentile. Beyond logarithmic bounds in online learning. In *AISTATS*, 2012. to appear.

[22] Dan Simon. A game theory approach to constrained minimax state estimation. *IEEE Transactions on Signal Processing*, 54(2):405–412, 2006.

[23] Dan Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[24] Eiji Takimoto and Manfred K. Warmuth. The last-step minimax algorithm. In *ALT*, 2000.

[25] Nina Vaits and Koby Crammer. Re-adapting the regularization of weights for non-stationary regression. In *ALT*, 2011.

[26] Volodimir G. Vovk. Aggregating strategies. In *COLT*, 1990.

[27] Volodya Vovk. Competitive on-line statistics. *International Statistical Review*, 69, 2001.

[28] Bernard Widrow and Jr. Marcian E. Hoff. Adaptive switching circuits. 1960.

[29] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.